

# Package ‘naniar’

February 9, 2018

**Type** Package

**Title** Data Structures, Summaries, and Visualisations for Missing Data

**Version** 0.2.0

**Description** Missing values are ubiquitous in data and need to be explored and handled in the initial stages of analysis. 'naniar' provides data structures and functions that facilitate the plotting of missing values and examination of imputations. This allows missing data dependencies to be explored with minimal deviation from the common work patterns of 'ggplot2' and tidy data.

**License** MIT + file LICENSE

**LazyData** TRUE

**ByteCompile** TRUE

**Suggests** knitr, rmarkdown, testthat, rpart, rpart.plot, covr, gridExtra, wakefield, vdiff, here, simulation, imputeTS

**VignetteBuilder** knitr

**Depends** R (>= 3.1.2)

**Imports** dplyr, ggplot2, purrr, tidyr, tibble, magrittr, stats, visdat, purrrlyr, rlang, forcats, viridis, glue

**RoxygenNote** 6.0.1.9000

**Collate** 'add-cols.R' 'add-n-prop-miss.R' 'cast-shadows.R' 'data-oceanbuoys.R' 'data-pedestrian.R' 'data-riskfactors.R' 'legend-draw.R' 'geom-miss-point.R' 'gg-miss-case-cumsum.R' 'gg-miss-case.R' 'gg-miss-fct.R' 'gg-miss-span.R' 'gg-miss-var-cumsum.R' 'gg-miss-var.R' 'gg-miss-which.R' 'label-miss.R' 'miss-prop-pct-summary.R' 'miss-scan-count.R' 'miss-x-cumsum.R' 'miss-x-pct-prop.R' 'miss-x-run.R' 'miss-x-span.R' 'miss-x-summary.R' 'miss-x-table.R' 'n-prop-miss-complete-rows.R' 'n-prop-miss-complete.R' 'naniar-ggproto.R' 'naniar-package.R' 'replace-to-na.R' 'replace-with-na.R' 'scoped-replace-with-na.R' 'shadow-shifters.R' 'shadows.R' 'stat-miss-point.R' 'utils.R' 'where-na.R'

**URL** <https://github.com/njtierney/naniar>

**BugReports** <https://github.com/njtierney/naniar/issues>

**NeedsCompilation** no

**Author** Nicholas Tierney [aut, cre] (<<https://orcid.org/0000-0003-1460-8722>>),  
 Di Cook [aut] (<<https://orcid.org/0000-0002-3813-7155>>),  
 Miles McBain [aut] (<<https://orcid.org/0000-0003-2865-2548>>),  
 Colin Fay [aut] (<<https://orcid.org/0000-0001-7343-1846>>),  
 Jim Hester [ctb]

**Maintainer** Nicholas Tierney <[nicholas.tierney@gmail.com](mailto:nicholas.tierney@gmail.com)>

**Repository** CRAN

**Date/Publication** 2018-02-09 06:01:23 UTC

## R topics documented:

add_any_miss . . . . .	4
add_label_missings . . . . .	5
add_label_shadow . . . . .	5
add_miss_cluster . . . . .	6
add_n_miss . . . . .	7
add_prop_miss . . . . .	8
add_shadow . . . . .	9
add_shadow_shift . . . . .	10
add_span_counter . . . . .	10
all_row_complete . . . . .	11
all_row_miss . . . . .	12
any_row_miss . . . . .	12
as_shadow . . . . .	13
as_shadow.data.frame . . . . .	13
bind_shadow . . . . .	14
cast_shadow . . . . .	15
cast_shadow_shift . . . . .	16
cast_shadow_shift_label . . . . .	16
gather_shadow . . . . .	17
GeomMissPoint . . . . .	18
geom_miss_point . . . . .	18
gg_miss_case . . . . .	20
gg_miss_case_cumsum . . . . .	21
gg_miss_fct . . . . .	22
gg_miss_span . . . . .	22
gg_miss_var . . . . .	23
gg_miss_var_cumsum . . . . .	24
gg_miss_which . . . . .	25
group_by_fun . . . . .	26
label_missings . . . . .	26
label_miss_1d . . . . .	27
label_miss_2d . . . . .	28
label_shadow . . . . .	28

label_shadow_matrix . . . . .	29
miss_case_pct . . . . .	29
miss_case_prop . . . . .	30
miss_case_summary . . . . .	31
miss_case_table . . . . .	32
miss_prop_summary . . . . .	32
miss_scan_count . . . . .	33
miss_summary . . . . .	34
miss_var_pct . . . . .	35
miss_var_prop . . . . .	36
miss_var_run . . . . .	36
miss_var_span . . . . .	38
miss_var_summary . . . . .	39
miss_var_table . . . . .	40
naniar . . . . .	40
n_complete . . . . .	41
n_complete_row . . . . .	42
n_miss . . . . .	42
n_miss_row . . . . .	43
oceanbuoys . . . . .	44
pct_complete . . . . .	45
pct_miss . . . . .	46
pedestrian . . . . .	47
prop_complete . . . . .	48
prop_complete_row . . . . .	48
prop_miss . . . . .	49
prop_miss_row . . . . .	50
replace_to_na . . . . .	50
replace_with_na . . . . .	51
replace_with_na_all . . . . .	52
replace_with_na_at . . . . .	53
replace_with_na_if . . . . .	54
riskfactors . . . . .	55
shadow_shift . . . . .	58
stat_miss_point . . . . .	59
test_if_dataframe . . . . .	60
test_if_missing . . . . .	60
test_if_null . . . . .	61
where_na . . . . .	62
which_na . . . . .	62
which_var_na . . . . .	63

---

 add\_any\_miss

*Add a column describing presence of any missing values*


---

### Description

This adds a column named "any\_miss" (by default) that describes whether there are any missings in all of the variables (default), or whether any of the specified columns, specified using variables names or dplyr verbs, starts\_with, contains, ends\_with, etc. By default the added column will be called "any\_miss\_all", if no variables are specified, otherwise, if variables are specified, the label will be "any\_miss\_vars" to indicate that not all variables have been used to create the labels.

### Usage

```
add_any_miss(data, ..., label = "any_miss")
```

### Arguments

data	data.frame
...	Variable names to use instead of the whole dataset. By default this looks at the whole dataset. Otherwise, this is one or more unquoted expressions separated by commas. These also respect the dplyr verbs starts_with, contains, ends_with, etc. By default will add "_all" to the label if left blank, otherwise will add "_vars" to distinguish that it has not been used on all of the variables.
label	label for the column, defaults to "any_miss". By default if no additional variables are listed the label col is "any_miss_all", otherwise it is "any_miss_vars", if variables are specified.

### Details

By default the prefix "any\_miss" is used, but this can be changed in the label argument.

### Value

data.frame with data and the column labelling whether that row (for those variables) has any missing values - indicated by "missing" and "complete".

### See Also

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

### Examples

```
airquality %>% add_any_miss()
airquality %>% add_any_miss(Ozone)
airquality %>% add_any_miss(Ozone, Solar.R)
```

---

add\_label\_missings      *Add a column describing if there are any missings in the dataset*

---

**Description**

Add a column describing if there are any missings in the dataset

**Usage**

```
add_label_missings(data)
```

**Arguments**

data                      data.frame

**Value**

data.frame with a column "any\_missing" that is either "Not Missing" or "Missing" for the purposes of plotting / exploration / nice print methods

**See Also**

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

**Examples**

```
airquality %>% add_label_missings()
```

---

add\_label\_shadow      *Add a column describing whether there is a shadow*

---

**Description**

Instead of focussing on labelling whether there are missings, we instead focus on whether there have been any shadows created. This can be useful when data has been imputed and you need to determine which rows contained missing values when the shadow was bound to the dataset.

**Usage**

```
add_label_shadow(data)
```

**Arguments**

data                      data.frame

**Value**

data.frame with a column, "any\_missing", which describes whether or not there are any rows that have a shadow value.

**See Also**

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

**Examples**

```
airquality %>%
  add_shadow(Ozone, Solar.R) %>%
  add_label_shadow()
```

---

add_miss_cluster	<i>Add a column that tells us which "missingness cluster" a row belongs to</i>
------------------	--

---

**Description**

A way to extract the cluster of missingness that a group belongs to. For example, if you use `vis_miss(airquality, cluster = TRUE)`, you can see some clustering in the data, but you do not have a way to identify the cluster. Future work will incorporate the `seriation` package to allow for better control over the clustering from the user.

**Usage**

```
add_miss_cluster(data, cluster_method = "mcquitty", n_clusters = 2)
```

**Arguments**

data	a dataframe
cluster_method	character vector of the agglomeration method to use, the default is "mcquitty". Options are taken from <code>stats::hclust</code> helpfile, and options include: "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
n_clusters	numeric the number of clusters you expect. Defaults to 2.

**See Also**

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

## Examples

```
add_miss_cluster(airquality)
add_miss_cluster(airquality, cluster_method = "ward.D")
add_miss_cluster(airquality, cluster_method = "ward.D", n_clusters = 3)
add_miss_cluster(airquality, n_clusters = 3)
```

---

add_n_miss	<i>Add column containing number of missing data values</i>
------------	--

---

## Description

It can be useful when doing data analysis to add the number of missing data points into your dataframe. `add_n_miss` adds a column named "n\_miss", which contains the number of missing values in that row.

## Usage

```
add_n_miss(data, ..., label = "n_miss")
```

## Arguments

<code>data</code>	a dataframe
<code>...</code>	Variable names to use instead of the whole dataset. By default this looks at the whole dataset. Otherwise, this is one or more unquoted expressions separated by commas. These also respect the dplyr verbs <code>starts_with</code> , <code>contains</code> , <code>ends_with</code> , etc. By default will add "_all" to the label if left blank, otherwise will add "_vars" to distinguish that it has not been used on all of the variables.
<code>label</code>	character default is "n_miss".

## Value

a dataframe

## See Also

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

## Examples

```
airquality %>% add_n_miss()
airquality %>% add_n_miss(Ozone, Solar.R)
airquality %>% add_n_miss(dplyr::contains("o"))
```

---

add_prop_miss	<i>Add column containing proportion of missing data values</i>
---------------	--

---

### Description

It can be useful when doing data analysis to add the proportion of missing data values into your dataframe. `add_prop_miss` adds a column named "prop\_miss", which contains the proportion of missing values in that row. You can specify the variables that you would like to show the missingness for.

### Usage

```
add_prop_miss(data, ..., label = "prop_miss")
```

### Arguments

<code>data</code>	a dataframe
<code>...</code>	Variable names to use instead of the whole dataset. By default this looks at the whole dataset. Otherwise, this is one or more unquoted expressions separated by commas. These also respect the dplyr verbs <code>starts_with</code> , <code>contains</code> , <code>ends_with</code> , etc. By default will add "_all" to the label if left blank, otherwise will add "_vars" to distinguish that it has not been used on all of the variables.
<code>label</code>	character string of what you need to name variable

### Value

a dataframe

### See Also

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

### Examples

```
airquality %>% add_prop_miss()

airquality %>% add_prop_miss(Solar.R)

airquality %>% add_prop_miss(Solar.R, Ozone)

airquality %>% add_prop_miss(Solar.R, Ozone, label = "testing")

# this can be applied to model the proportion of missing data
# as in Tierney et al bmjopen.bmj.com/content/5/6/e007450.full
library(rpart)
library(rpart.plot)
```



```
airquality %>%  
  add_prop_miss() %>%  
  rpart(prop_miss_all ~ ., data = .) %>%  
  prp(type = 4,  
      extra = 101,  
      prefix = "prop_miss = ")
```

---

add\_shadow

*Add a shadow column to dataframe*

---

## Description

As an alternative to `bind_shadow()`, you can add specific individual shadow columns to a dataset. These also respect the dplyr verbs `starts_with`, `contains`, `ends_with`, etc.

## Usage

```
add_shadow(data, ...)
```

## Arguments

<code>data</code>	<code>data.frame</code>
<code>...</code>	One or more unquoted variable names, separated by commas. These also respect the dplyr verbs <code>starts_with</code> , <code>contains</code> , <code>ends_with</code> , etc.

## Value

`data.frame`

## See Also

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

## Examples

```
airquality %>% add_shadow(Ozone)  
airquality %>% add_shadow(Ozone, Solar.R)
```

---

add\_shadow\_shift      *Add a shadow shifted column to a dataset*

---

### Description

Shadow shift missing values using only the selected variables in a dataset, by specifying variable names or use dplyr vars and dplyr verbs starts\_with, contains, ends\_with, etc.

### Usage

```
add_shadow_shift(data, ..., suffix = "shift")
```

### Arguments

data	data.frame
...	One or more unquoted variable names separated by commas. These also respect the dplyr verbs starts_with, contains, ends_with, etc.
suffix	suffix to add to variable, defaults to "shift"

### Value

data with the added variable shifted named as var\_suffix

### See Also

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

### Examples

```
pedestrian %>% add_shadow_shift(hourly_counts)
```

```
airquality %>% add_shadow_shift(Ozone, Solar.R)
```

---

add\_span\_counter      *Add a counter variable for a span of dataframe*

---

### Description

Adds a variable, span\_counter to a dataframe. Used internally to facilitate counting of missing values over a given span.

**Usage**

```
add_span_counter(data, span_size)
```

**Arguments**

data	data.frame
span_size	integer

**Value**

data.frame with extra variable "span\_counter".

**Examples**

```
## Not run:  
add_span_counter(pedestrian, span_size = 100)  
  
## End(Not run)
```

---

<code>all_row_complete</code>	<i>Helper function to determine whether all rows are complete</i>
-------------------------------	---

---

**Description**

Helper function to determine whether all rows are complete

**Usage**

```
all_row_complete(x)
```

**Arguments**

x	a vector
---	----------

**Value**

logical vector

---

`all_row_miss`*Helper function to determine whether all rows are missing*

---

**Description**

Helper function to determine whether all rows are missing

**Usage**

```
all_row_miss(x)
```

**Arguments**

`x` a vector

**Value**

logical vector

---

`any_row_miss`*Helper function to determine whether there are any missings*

---

**Description**

Helper function to determine whether there are any missings

**Usage**

```
any_row_miss(x)
```

**Arguments**

`x` a vector

**Value**

logical vector TRUE = missing FALSE = complete

---

as_shadow	<i>Create shadows</i>
-----------	-----------------------

---

**Description**

Representing missing data structure is achieved using the shadow matrix, introduced in [Swayne and Buja](#). The shadow matrix is the same dimension as the data, and consists of binary indicators of missingness of data values, where missing is represented as "NA", and not missing is represented as "!NA". Although these may be represented as 1 and 0, respectively.

**Usage**

```
as_shadow(data, ...)
```

**Arguments**

data	dataframe
...	selected variables to use

**Value**

appended shadow with column names

---

as_shadow.data.frame	<i>Create shadow data</i>
----------------------	---------------------------

---

**Description**

Return a tibble in shadow matrix form, where the variables are the same but have a suffix `_NA` attached to distinguish them.

**Usage**

```
## S3 method for class 'data.frame'
as_shadow(data, ...)
```

**Arguments**

data	dataframe
...	selected variables to use

**Examples**

```
as_shadow(airquality)
```

---

`bind_shadow`*Bind a shadow dataframe to original data*

---

## Description

Binding a shadow matrix to a regular dataframe helps visualise and work with missing data.

## Usage

```
bind_shadow(data, only_miss = FALSE)
```

## Arguments

<code>data</code>	a dataframe
<code>only_miss</code>	logical - if FALSE (default) it will bind a dataframe with all of the variables duplicated with their shadow. Setting this to TRUE will bind variables only those variables that contain missing values. See the examples for more details.

## Value

data with the added variable shifted and the suffix `_NA`

## Examples

```
bind_shadow(airquality)

# bind only the variables that contain missing values
bind_shadow(airquality, only_miss = TRUE)

aq_shadow <- bind_shadow(airquality)

# explore missing data visually
library(ggplot2)

# using the bounded shadow to visualise Ozone according to whether Solar
# Radiation is missing or not.

ggplot(data = aq_shadow,
        aes(x = Ozone)) +
  geom_histogram() +
  facet_wrap(~Solar.R_NA,
            ncol = 1)
```

---

cast_shadow	<i>Add a shadow column to a dataset</i>
-------------	---

---

## Description

Casting a shadow shifted column performs the equivalent pattern to data that makes it easy to perform certain visualisations, in line with the principle that the user should have a way to flexibly return data formats containing information about the missing data. It forms the base building block for the functions `cast_shadow_shift`, and `cast_shadow_shift_label`. It also respects the dplyr verbs `starts_with`, `contains`, `ends_with`, etc. to select variables.

## Usage

```
cast_shadow(data, ...)
```

## Arguments

<code>data</code>	<code>data.frame</code>
<code>...</code>	One or more unquoted variable names separated by commas. These respect the dplyr verbs <code>starts_with</code> , <code>contains</code> , <code>ends_with</code> , etc.

## Value

data with the added variable shifted and the suffix `_NA`

## See Also

[cast\\_shadow\\_shift\(\)](#), [cast\\_shadow\\_shift\\_label\(\)](#) [bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#)  
[add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#)

## Examples

```
airquality %>% cast_shadow(Ozone)
airquality %>% cast_shadow(Ozone, Solar.R)
library(ggplot2)
library(magrittr)
airquality %>%
  cast_shadow(Ozone, Solar.R) %>%
  ggplot(aes(x = Ozone,
             colour = Solar.R_NA)) +
  geom_density()
```

---

cast\_shadow\_shift      *Add a shadow and a shadow\_shift column to a dataset*

---

### Description

Shift the values and add a shadow column. It also respects the dplyr verbs starts\_with, contains, ends\_with, etc.

### Usage

```
cast_shadow_shift(data, ...)
```

### Arguments

data	data.frame
...	One or more unquoted variable names separated by commas. These respect the dplyr verbs starts_with, contains, ends_with, etc.

### Value

data.frame with the shadow and shadow\_shift vars

### See Also

[cast\\_shadow\\_shift\(\)](#), [cast\\_shadow\\_shift\\_label\(\)](#) [bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#)  
[add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#)

### Examples

```
airquality %>% cast_shadow_shift(Ozone)
airquality %>% cast_shadow_shift(Ozone,Temp)

airquality %>% cast_shadow_shift(dplyr::contains("o"))
```

---

cast\_shadow\_shift\_label  
*Add a shadow column and a shadow shifted column to a dataset*

---

### Description

Shift the values, add shadow, add missing label

### Usage

```
cast_shadow_shift_label(data, ...)
```



**Arguments**

data            data.frame

...            One or more unquoted expressions separated by commas. These also respect the dplyr verbs "starts\_with", "contains", "ends\_with", etc.

**Value**

data.frame with the shadow and shadow\_shift vars, and missing labels

**See Also**

[cast\\_shadow\\_shift\(\)](#), [cast\\_shadow\\_shift\\_label\(\)](#) [bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#)  
[add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#)

**Examples**

```
airquality %>% cast_shadow_shift_label(Ozone)
airquality %>% cast_shadow_shift_label(Ozone, Solar.R)

# replicate the plot generated by geom_miss_point()

library(ggplot2)

airquality %>%
  cast_shadow_shift_label(Ozone, Solar.R) %>%
  ggplot(aes(x = Ozone_shift,
            y = Solar.R_shift,
            colour = any_missing)) +
  geom_point()
```

---

gather\_shadow

*Long form representation of a shadow matrix*


---

**Description**

gather\_shadow is a long-form representation of binding the shadow matrix to your data, producing variables named case, variable, and missing, where missing contains the missing value representation.

**Usage**

```
gather_shadow(data)
```

**Arguments**

data            a dataframe

**Value**

dataframe in long, format, containing information about the missings

**Examples**

```
gather_shadow(airquality)
```

---

GeomMissPoint	<i>naniar-ggproto</i>
---------------	-----------------------

---

**Description**

These are the stat and geom overrides using ggproto from ggplot2 that make naniar work.

**Usage**

```
StatMissPoint
```

**Format**

An object of class StatMissPoint (inherits from Stat, ggproto) of length 6.

---

geom_miss_point	<i>geom_miss_point</i>
-----------------	------------------------

---

**Description**

`geom_miss_point` provides a way to transform and plot missing values in ggplot2. To do so it uses methods from ggobi to display missing data points 10% below the minimum value, so that the values can be seen on the same axis.

**Usage**

```
geom_miss_point(mapping = NULL, data = NULL, position = "identity",
  colour = ..missing.., na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)
```

**Arguments**

mapping	Set of aesthetic mappings created by <code>ggplot2::aes()</code> or <code>ggplot2::aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), is combined with the default mapping at the top level of the plot. You only need to supply mapping if there isn't a mapping defined for the plot.
data	A data frame. If specified, overrides the default data frame defined at the top level of the plot.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
colour	the colour chosen for the aesthetic
na.rm	If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders.
...	other arguments passed on to <code>ggplot2::layer()</code> . There are three types of arguments you can use here: <ul style="list-style-type: none"> <li>• Aesthetics: to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code>.</li> <li>• Other arguments to the layer, for example you override the default stat associated with the layer.</li> <li>• Other arguments passed on to the stat.</li> </ul>

**Details**

Plot Missing Data Points

**Note**

Warning message if `na.rm = T` is supplied.

**See Also**

`[gg_miss_case()]``[gg_miss_case_cumsum()]``[gg_miss_fct()]``[gg_miss_span()]``[gg_miss_var()]``[gg_miss_var_cumsum()]``[gg_`

**Examples**

```
library(ggplot2)

# using regular geom_point()
ggplot(airquality,
       aes(x = Ozone,
          y = Solar.R)) +
  geom_point()
```

```
# using geom_miss_point()
ggplot(airquality,
       aes(x = Ozone,
           y = Solar.R)) +
  geom_miss_point()

# using facets

ggplot(airquality,
       aes(x = Ozone,
           y = Solar.R)) +
  geom_miss_point() +
  facet_wrap(~Month)
```

---

gg_miss_case	<i>Plot the number of missings per case (row)</i>
--------------	---

---

## Description

This is a visual analogue to `miss_case_summary`. It draws a ggplot of the number of missings in each case (row). A default minimal theme is used, which can be customised as normal for ggplot.

## Usage

```
gg_miss_case(x, facet, order_cases = FALSE)
```

## Arguments

x	data.frame
facet	(optional) a single bare variable name, if you want to create a faceted plot.
order_cases	logical Order the rows by missingness (default is FALSE - no order).

## Value

a ggplot object depicting the number of missings in a given case.

## See Also

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_span\(\)](#) [gg\\_miss\\_var\(\)](#)  
[gg\\_miss\\_var\\_cumsum\(\)](#) [gg\\_miss\\_which\(\)](#)

## Examples

```
gg_miss_case(airquality)
library(ggplot2)
gg_miss_case(airquality) + labs(x = "Number of Cases")
gg_miss_case(airquality, order_cases = TRUE)
gg_miss_case(airquality, facet = Month)
gg_miss_case(airquality, facet = Month, order_cases = TRUE)
```

---

gg\_miss\_case\_cumsum *Plot of cumulative sum of missing for cases*

---

## Description

A plot showing the cumulative sum of missing values for cases, reading the rows from the top to bottom. A default minimal theme is used, which can be customised as normal for ggplot.

## Usage

```
gg_miss_case_cumsum(x, breaks = 20)
```

## Arguments

x	a dataframe
breaks	the breaks for the x axis default is 20

## Value

a ggplot object depicting the number of missings

## See Also

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_span\(\)](#) [gg\\_miss\\_var\(\)](#) [gg\\_miss\\_var\\_cumsum\(\)](#)  
[gg\\_miss\\_which\(\)](#)

## Examples

```
gg_miss_case_cumsum(airquality)
library(ggplot2)
gg_miss_case_cumsum(riskfactors, breaks = 50) + theme_bw()
```

---

gg_miss_fct	<i>Plot the number of missings for each variable, broken down by a factor</i>
-------------	---

---

### Description

This function draws a ggplot plot of the number of missings in each column, broken down by a categorical variable from the dataset. A default minimal theme is used, which can be customised as normal for ggplot.

### Usage

```
gg_miss_fct(x, fct)
```

### Arguments

x	data.frame
fct	column containing the factor variable to visualise

### Value

ggplot object depicting the each variable.

### See Also

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_span\(\)](#) [gg\\_miss\\_var\(\)](#)  
[gg\\_miss\\_var\\_cumsum\(\)](#) [gg\\_miss\\_which\(\)](#)

### Examples

```
gg_miss_fct(x = riskfactors, fct = marital)
library(ggplot2)
gg_miss_fct(x = riskfactors, fct = marital) + labs(title = "NA in Risk Factors and Marital status")
```

---

gg_miss_span	<i>Plot the number of missings in a given repeating span</i>
--------------	--

---

### Description

gg\_miss\_span is a replacement function to `imputeTS::plotNA.distributionBar(tsNH4, breaksize = 100)`, which shows the number of missings in a given span, or breaksize. A default minimal theme is used, which can be customised as normal for ggplot.

**Usage**

```
gg_miss_span(data, var, span_every, facet)
```

**Arguments**

data	data.frame
var	a bare unquoted variable name from data.
span_every	integer describing the length of the span to be explored
facet	(optional) a single bare variable name, if you want to create a faceted plot.

**Value**

ggplot2 showing the number of missings in a span (window, or breaksize)

**See Also**

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_var\(\)](#)  
[gg\\_miss\\_var\\_cumsum\(\)](#) [gg\\_miss\\_which\(\)](#)

**Examples**

```
miss_var_span(pedestrian, hourly_counts, span_every = 3000)
library(ggplot2)
gg_miss_span(pedestrian, hourly_counts, span_every = 3000)
gg_miss_span(pedestrian, hourly_counts, span_every = 3000, facet = sensor_name)
# works with the rest of ggplot
gg_miss_span(pedestrian, hourly_counts, span_every = 3000) + labs(x = "custom")
gg_miss_span(pedestrian, hourly_counts, span_every = 3000) + theme_dark()

gg_miss_span(pedestrian, hourly_counts, span_every = 3000, facet = sensor_name)
```

---

gg\_miss\_var

*Plot the number of missings for each variable*

---

**Description**

This is a visual analogue to `miss_var_summary`. It draws a ggplot of the number of missings in each variable, ordered to show which variables have the most missing data. A default minimal theme is used, which can be customised as normal for ggplot.

**Usage**

```
gg_miss_var(x, facet, show_pct = FALSE)
```

**Arguments**

x	a dataframe
facet	(optional) bare variable name, if you want to create a faceted plot.
show_pct	logical shows the number of missings (default), but if set to TRUE, it will display the proportion of missings.

**Value**

a ggplot object depicting the number of missings in a given column

**See Also**

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_span\(\)](#)  
[gg\\_miss\\_var\(\)](#) [gg\\_miss\\_var\\_cumsum\(\)](#) [gg\\_miss\\_which\(\)](#)

**Examples**

```
gg_miss_var(airquality)
library(ggplot2)
gg_miss_var(airquality) + labs(y = "Look at all the missing ones")
gg_miss_var(airquality, Month)
gg_miss_var(airquality, Month, TRUE)
```

---

`gg_miss_var_cumsum`     *Plot of cumulative sum of missing value for each variable*

---

**Description**

A plot showing the cumulative sum of missing values for each variable, reading columns from the left to the right of the initial dataframe. A default minimal theme is used, which can be customised as normal for ggplot.

**Usage**

```
gg_miss_var_cumsum(x)
```

**Arguments**

x	a data.frame
---	--------------

**Value**

a ggplot object showing the cumulative sum of missings over the variables



**See Also**

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_span\(\)](#)  
[gg\\_miss\\_var\(\)](#) [gg\\_miss\\_which\(\)](#)

**Examples**

```
gg_miss_var_cumsum(airquality)
```

---

`gg_miss_which` *Plot which variables contain a missing value*

---

**Description**

This plot produces a set of rectangles indicating whether there is a missing element in a column or not. A default minimal theme is used, which can be customised as normal for ggplot.

**Usage**

```
gg_miss_which(x)
```

**Arguments**

`x` a dataframe

**Value**

a ggplot object of which variables contains missing values

**See Also**

[geom\\_miss\\_point\(\)](#) [gg\\_miss\\_case\(\)](#) [gg\\_miss\\_case\\_cumsum](#) [gg\\_miss\\_fct\(\)](#) [gg\\_miss\\_span\(\)](#)  
[gg\\_miss\\_var\(\)](#) [gg\\_miss\\_var\\_cumsum\(\)](#) [gg\\_miss\\_which\(\)](#)

**Examples**

```
gg_miss_which(airquality)  
library(ggplot2)
```

---

group_by_fun	<i>Group By Helper</i>
--------------	------------------------

---

**Description**

This is a wrapper to facilitate the grouped\_df S3 method.

**Usage**

```
group_by_fun(data, .fun, ...)
```

**Arguments**

data	data.frame, which will be grouped
.fun	a function to apply
...	additional arguments to be passed to map

**Value**

a dataframe with the function applied to each group

**Examples**

```
## Not run:
miss_case_table.grouped_df <- function(data){
  group_by_fun(data, .fun = miss_case_table)
}
airquality %>%
  group_by(Month) %>%
  miss_case_table()

## End(Not run)
```

---

label_missings	<i>Is there a missing value in the row of a dataframe?</i>
----------------	--

---

**Description**

Creates a character vector describing presence/absence of missing values

**Usage**

```
label_missings(data)
```

**Arguments**

data                    a dataframe or set of vectors of the same length

**Value**

character vector of "Missing" and "Not Missing".

**See Also**

[bind\\_shadow\(\)](#) [add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#) [add\\_miss\\_cluster\(\)](#)  
[add\\_n\\_miss\(\)](#) [add\\_prop\\_miss\(\)](#) [add\\_shadow\\_shift\(\)](#) [cast\\_shadow\(\)](#)

**Examples**

```
label_missings(airquality)

library(dplyr)

airquality %>% mutate(is_missing = label_missings(airquality))
```

---

label_miss_1d	<i>Label a missing from one column</i>
---------------	--

---

**Description**

Label whether a value is missing in a row of one columns.

**Usage**

```
label_miss_1d(x1)
```

**Arguments**

x1                    a variable of a dataframe

**Value**

a vector indicating whether any of these rows had missing values

**Note**

can we generalise label\_miss to work for any number of variables?

**See Also**

[add\\_any\\_miss\(\)](#) [add\\_label\\_missings\(\)](#) [add\\_label\\_shadow\(\)](#)

**Examples**

```
label_miss_1d(airquality$Ozone)
```

---

label_miss_2d	<i>label_miss_2d</i>
---------------	----------------------

---

**Description**

Label whether a value is missing in either row of two columns.

**Usage**

```
label_miss_2d(x1, x2)
```

**Arguments**

x1	a variable of a dataframe
x2	another variable of a dataframe

**Value**

a vector indicating whether any of these rows had missing values

**Examples**

```
label_miss_2d(airquality$Ozone, airquality$Solar.R)
```

---

label_shadow	<i>Label shadow values as missing or not missing</i>
--------------	--

---

**Description**

Powers add\_label\_shadow. For the moment it is an internal function.

**Usage**

```
label_shadow(data)
```

**Arguments**

data	data.frame
------	------------

**Value**

"Missing" or "Not Missing"

---

label\_shadow\_matrix     *Give NAs a more meaningful label*

---

**Description**

Returns a binary factor of !NA and NA, where !NA indicates a datum that is not missing, and NA indicates missingness. This function is what powers the factor levels in as\_shadow().

**Usage**

```
label_shadow_matrix(x)
```

**Arguments**

x                    a vector

**Value**

a vector of factors containing the labels "!NA" for Not missing and "NA" for missing.

**See Also**

as\_shadow

**Examples**

```
## Not run:  
label_shadow_matrix(airquality$Ozone)  
  
## End(Not run)
```

---

miss\_case\_pct             *Percentage of cases that contain a missing values.*

---

**Description**

Calculate the percentage of cases (rows) that contain a missing value.

**Usage**

```
miss_case_pct(data)
```

**Arguments**

data                    a dataframe

**Value**

numeric the percentage of cases that contain a missing value

**See Also**

[miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#) [miss\\_summary](#)  
[miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_case_pct(airquality)
```

---

miss_case_prop	<i>Proportion of cases that contain a missing values.</i>
----------------	---

---

**Description**

Calculate the proportion of cases (rows) that contain a missing value.

**Usage**

```
miss_case_prop(data)
```

**Arguments**

data            a dataframe

**Value**

numeric the proportion of cases that contain a missing value

**See Also**

[miss\\_case\\_pct](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#) [miss\\_summary](#)  
[miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_case_prop(airquality)
```

---

miss_case_summary	<i>Summarise the missingness in each case</i>
-------------------	---

---

## Description

Return for each case the number and percent of missing values, ordered by the most number of missings.

## Usage

```
miss_case_summary(data, order = FALSE, ...)
```

## Arguments

data	a data.frame
order	a logical indicating whether or not to order the result by n_miss. TRUE orders from largest to smallest n_miss, and FALSE orders by order provided by the data.
...	extra arguments

## Value

a tibble of the percent of missing data in each case.

## See Also

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_table](#) [miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

## Examples

```
# works with group_by from dplyr
library(dplyr)
airquality %>%
  group_by(Month) %>%
  miss_case_summary()

miss_case_summary(airquality)
```

---

miss_case_table	<i>Tabulate missings in cases.</i>
-----------------	------------------------------------

---

**Description**

Provide a tidy table of the number of cases with 0, 1, 2, up to n, missing values and the proportion of the number of cases those cases make up.

**Usage**

```
miss_case_table(data)
```

**Arguments**

data            a dataframe

**Value**

a dataframe

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_case_table(airquality)
library(dplyr)
airquality %>%
  group_by(Month) %>%
  miss_case_table()
```

---

miss_prop_summary	<i>Proportions of missings in data, variables, and cases.</i>
-------------------	---

---

**Description**

Return missing data info about the dataframe, the variables, and the cases. Specifically, returning how many elements in a dataframe contain a missing value, how many elements in a variable contain a missing value, and how many elements in a case contain a missing.

**Usage**

```
miss_prop_summary(data)
```



**Arguments**

data            a dataframe

**Value**

a dataframe

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#) [miss\\_scan\\_count](#) [miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_prop_summary(airquality)
library(dplyr)
airquality %>% group_by(Month) %>% miss_prop_summary()
```

---

miss\_scan\_count

*Search and present different kinds of missing values*

---

**Description**

Searching for different kinds of missing values is really annoying. If you have values like -99 in your data, when they shouldn't be there, or they should be encoded as missing, it can be difficult to ascertain if they are there, and if so, where they are. `miss_scan_count` makes it easier for users to search for particular occurrences of these values across their variables.

**Usage**

```
miss_scan_count(data, search)
```

**Arguments**

data            data  
search          values to search for

**Value**

a dataframe of the occurrences of the values you searched for

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#) [miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
dat_ms <- tibble::tribble(~x, ~y, ~z,
  1, "A", -100,
  3, "N/A", -99,
  NA, NA, -98,
  -99, "E", -101,
  -98, "F", -1)

miss_scan_count(dat_ms, -99)
miss_scan_count(dat_ms, c(-99, -98))
miss_scan_count(dat_ms, c("-99", "-98", "N/A"))
```

---

miss\_summary

*Collate summary measures from naniar into one tibble*


---

**Description**

miss\_summary performs all of the missing data helper summaries and puts them into lists within a tibble

**Usage**

```
miss_summary(data, order = FALSE)
```

**Arguments**

data	a dataframe
order	whether or not to order the result by n_miss
...	extra arguments

**Value**

a tibble of missing data summaries

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
s_miss <- miss_summary(airquality)
s_miss$miss_df_prop
s_miss$miss_case_table
s_miss$miss_var_summary
# etc, etc, etc.

library(dplyr)
s_miss_group <- group_by(airquality, Month) %>% miss_summary()
s_miss_group$miss_df_prop
s_miss_group$miss_case_table
# etc, etc, etc.
```

---

miss_var_pct	<i>Percentage of variables containing missings</i>
--------------	--

---

**Description**

Calculate the percentage of variables that contain a single missing value.

**Usage**

```
miss_var_pct(data)
```

**Arguments**

data            a dataframe

**Value**

numeric the percent of variables that contain missing data

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_](#)  
[table](#)

**Examples**

```
miss_var_pct(riskfactors)
miss_var_pct(oceanbuoys)
```

---

miss_var_prop	<i>Proportion of variables containing missings</i>
---------------	--

---

**Description**

Calculate the proportion of variables that contain a single missing value.

**Usage**

```
miss_var_prop(data)
```

**Arguments**

data            a dataframe

**Value**

numeric the proportion of variables that contain missing data

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_var_prop(riskfactors)
miss_var_prop(oceanbuoys)
```

---

miss_var_run	<i>Find the number of missing and complete values in a single run</i>
--------------	---

---

**Description**

It is useful to find the number of missing values that occur in a single run. The function, `miss_var_run()`, returns a dataframe with the column names "run\_length" and "is\_na", which describe the length of the run, and whether that run describes a missing value.

**Usage**

```
miss_var_run(data, var)
```

**Arguments**

data	data.frame
var	a bare variable name

**Value**

dataframe with column names "run\_length" and "is\_na", which describe the length of the run, and whether that run describes a missing value.

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

**Examples**

```
miss_var_run(pedestrian, hourly_counts)

library(dplyr)

# find the number of runs missing/complete for each month

pedestrian %>%
  group_by(month) %>%
  miss_var_run(hourly_counts)

library(ggplot2)

# explore the number of missings in a given run
miss_var_run(pedestrian, hourly_counts) %>%
  filter(is_na == "missing") %>%
  count(run_length) %>%
  ggplot(aes(x = run_length,
             y = n)) +
  geom_col()

# look at the number of missing values and the run length of these.
miss_var_run(pedestrian, hourly_counts) %>%
  ggplot(aes(x = is_na,
             y = run_length)) +
  geom_boxplot()

# using group_by
pedestrian %>%
  group_by(month) %>%
  miss_var_run(hourly_counts)
```

---

miss_var_span	<i>Summarise the number of missings for a given repeating span on a variable</i>
---------------	--

---

### Description

To summarise the missing values in a time series object it can be useful to calculate the number of missing values in a given time period. `miss_var_span` takes a `data.frame` object, a variable, and a `span_every` argument and returns a dataframe containing the number of missing values within each span.

### Usage

```
miss_var_span(data, var, span_every)
```

### Arguments

<code>data</code>	<code>data.frame</code>
<code>var</code>	bare unquoted variable name of interest.
<code>span_every</code>	integer describing the length of the span to be explored

### Value

dataframe with variables `n_miss`, `n_complete`, `prop_miss`, and `prop_complete`, which describe the number, or proportion of missing or complete values within that given time span.

### See Also

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_summary](#) [miss\\_var\\_table](#)

### Examples

```
miss_var_span(data = pedestrian,  
              var = hourly_counts,  
              span_every = 168)  
  
library(dplyr)  
pedestrian %>%  
  group_by(month) %>%  
  miss_var_span(var = hourly_counts,  
               span_every = 168)
```

---

miss_var_summary	<i>Summarise the missingness in each variable</i>
------------------	---

---

### Description

Provide a summary for each variable of the number and percent missings, ordering by the most missings in each variable.

### Usage

```
miss_var_summary(data, order = FALSE, ...)
```

### Arguments

data	a data.frame
order	a logical indicating whether or not to order the result by n_miss. TRUE orders from largest to smallest n_miss, and FALSE orders by order provided by the data.
...	extra arguments

### Value

a tibble of the percent of missing data in each variable

### See Also

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_table](#)

### Examples

```
miss_var_summary(airquality)
miss_var_summary(oceanbuoys, order = TRUE)

# works with group_by from dplyr
library(dplyr)
airquality %>%
  group_by(Month) %>%
  miss_var_summary()
```

---

miss_var_table	<i>Tabulate the missings in the variables</i>
----------------	---

---

**Description**

Provide a tidy table of the number of variables with 0, 1, 2, up to n, missing values and the proportion of the number of variables those variables make up.

**Usage**

```
miss_var_table(data)
```

**Arguments**

data            a dataframe

**Value**

a dataframe

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#)  
[miss\\_var\\_table](#)

**Examples**

```
miss_var_table(airquality)

library(dplyr)
airquality %>%
  group_by(Month) %>%
  miss_var_table()
```

---

naniar	<i>naniar</i>
--------	---------------

---

**Description**

naniar is a package to make it easier to summarise and handle missing values in R. It strives to do this in a way that is as consistent with tidyverse principles as possible.



**See Also**

add\_any\_miss add\_label\_missings add\_label\_shadow add\_miss\_cluster add\_n\_miss add\_prop\_miss add\_shadow add\_shadow\_shift as\_shadow bind\_shadow cast\_shadow cast\_shadow\_shift cast\_shadow\_shift\_label draw\_key\_missing\_point gather\_shadow geom\_miss\_point gg\_miss\_case gg\_miss\_case\_cumsum gg\_miss\_fct gg\_miss\_span gg\_miss\_var gg\_miss\_var\_cumsum gg\_miss\_which label\_miss\_1d label\_miss\_2d label\_missings miss\_case\_pct miss\_case\_prop miss\_case\_summary miss\_case\_table miss\_prop\_summary miss\_scan\_count miss\_summary miss\_var\_pct miss\_var\_prop miss\_var\_run miss\_var\_span miss\_var\_summary miss\_var\_table n\_complete n\_complete\_row n\_miss n\_miss\_row pct\_complete pct\_miss prop\_complete prop\_complete\_row prop\_miss prop\_miss\_row replace\_to\_na replace\_with\_na replace\_with\_na\_all replace\_with\_na\_at replace\_with\_na\_if shadow\_shift stat\_miss\_point vis\_miss where\_na

---

n_complete	<i>Return the number of complete values</i>
------------	---

---

**Description**

A complement to ‘n\_miss‘

**Usage**

```
n_complete(x)
```

**Arguments**

x                    a vector

**Value**

numeric number of complete values

**Examples**

```
n_complete(airquality)
n_complete(airquality$Ozone)
```

---

n_complete_row	<i>Return a vector of the number of complete values in each row</i>
----------------	---

---

**Description**

Substitute for `rowSums(!is.na(data))` but it also checks if input is `NULL` or is a dataframe

**Usage**

```
n_complete_row(data)
```

**Arguments**

data            a dataframe

**Value**

numeric vector of the number of complete values in each row

**See Also**

`miss_case_pct` `miss_case_prop` `miss_prop_summary()` `miss_case_summary` `miss_case_table`  
`miss_summary` `miss_var_pct` `miss_var_prop` `miss_var_run` `miss_var_span` `miss_var_summary`  
`miss_var_table` `n_complete` `n_complete_row` `n_miss` `n_miss_row` `pct_complete` `pct_miss`  
`prop_complete` `prop_complete_row` `prop_miss`

**Examples**

```
n_complete_row(airquality)
n_complete_row(pedestrian)
```

---

n_miss	<i>Return the number of missing values</i>
--------	--

---

**Description**

Substitute for `sum(is.na(data))`

**Usage**

```
n_miss(x)
```

**Arguments**

x                a vector

**Value**

numeric the number of missing values

**Examples**

```
n_miss(airquality)
n_miss(airquality$Ozone)
```

---

n_miss_row	<i>Return a vector of the number of missing values in each row</i>
------------	--

---

**Description**

Substitute for `rowSums(is.na(data))`, but it also checks if input is NULL or is a dataframe

**Usage**

```
n_miss_row(data)
```

**Arguments**

data            a dataframe

**Value**

numeric vector of the number of missing values in each row

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#)  
[miss\\_var\\_table](#) [n\\_complete](#) [n\\_complete\\_row](#) [n\\_miss](#) [n\\_miss\\_row](#) [pct\\_complete](#) [pct\\_miss](#)  
[prop\\_complete](#) [prop\\_complete\\_row](#) [prop\\_miss](#)

**Examples**

```
n_miss_row(airquality)
n_miss_row(pedestrian)
```

---

oceanbuoys

*West Pacific Tropical Atmosphere Ocean Data, 1993 & 1997.*

---

### Description

Real-time data from moored ocean buoys for improved detection, understanding and prediction of El Niño and La Niña. The data is collected by the Tropical Atmosphere Ocean project (<http://www.pmel.noaa.gov/tao/index.shtml>).

### Usage

```
data(oceanbuoys)
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 736 rows and 8 columns.

### Details

Format: a data frame with 736 observations on the following 8 variables.

`year` A numeric with levels 1993 1997.

`latitude` A numeric with levels -5 -2 0.

`longitude` A numeric with levels -110 -95.

`sea_temp_c` Sea surface temperature(degree Celsius), measured by the TAO buoys at one meter below the surface.

`air_temp_c` Air temperature(degree Celsius), measured by the TAO buoys three meters above the sea surface.

`humidity` Relative humidity( meters above the sea surface.

`wind_ew` The East-West wind vector components(M/s). TAO buoys measure the wind speed and direction four meters above the sea surface. If it is positive, the East-West component of the wind is blowing towards the East. If it is negative, this component is blowing towards the West.

`wind_ns` The North-South wind vector components(M/s). TAO buoys measure the wind speed and direction four meters above the sea surface. If it is positive, the North-South component of the wind is blowing towards the North. If it is negative, this component is blowing towards the South.

### Source

[http://www.pmel.noaa.gov/tao/data\\_deliv/deliv.html](http://www.pmel.noaa.gov/tao/data_deliv/deliv.html)

### See Also

`library(MissingDataGUI)` (data named "tao")

## Examples

```
# explore the missingness with vis_miss
library(naniar)

vis_miss(oceanbuoys)

# Look at the missingness in the variables
miss_var_summary(oceanbuoys)

# Look at the missingness in air temperature and humidity
library(ggplot2)
p <-
ggplot(oceanbuoys,
       aes(x = air_temp_c,
           y = humidity)) +
  geom_miss_point()

p

# for each year?
p + facet_wrap(~year)

# this shows that there are more missing values in humidity in 1993, and
# more air temperature missing values in 1997

# what if we explore the value of air temperature and humidity based on
# the missingness of each

oceanbuoys %>%
  bind_shadow() %>%
  ggplot(aes(x = air_temp_c,
            fill = humidity_NA)) +
  geom_histogram()

oceanbuoys %>%
  bind_shadow() %>%
  ggplot(aes(x = humidity,
            fill = air_temp_c_NA)) +
  geom_histogram()
```

---

pct\_complete

*Return the percent of complete values*

---

## Description

The complement to pct\_miss

**Usage**

```
pct_complete(x)
```

**Arguments**

x                    vector or data.frame

**Value**

numeric percent of complete values

**Examples**

```
pct_complete(airquality)
pct_complete(airquality$Ozone)
```

---

pct\_miss

*Return the percent of missing values*

---

**Description**

This is shorthand for `mean(is.na(x)) * 100`

**Usage**

```
pct_miss(x)
```

**Arguments**

x                    vector or data.frame

**Value**

numeric the percent of missing values in x

**Examples**

```
pct_miss(airquality)
pct_miss(airquality$Ozone)
```

---

pedestrian

*Pedestrian count information around Melbourne for 2016*

---

## Description

This dataset contains hourly counts of pedestrians from 4 sensors around Melbourne: Birrarung Marr, Bourke Street Mall, Flagstaff station, and Spencer St-Collins St (south), recorded from January 1st 2016 at 00:00:00 to December 31st 2016 at 23:00:00. The data is made free and publicly available from <https://data.melbourne.vic.gov.au/Transport-Movement/Pedestrian-volume-updated-monthly-b2ak-trbp>

## Usage

```
data(pedestrian)
```

## Format

A tibble with 37,700 rows and 9 variables:

**hourly\_counts** (integer) the number of pedestrians counted at that sensor at that time

**date\_time** (POSIXct, POSIXt) The time that the count was taken

**year** (integer) Year of record

**month** (factor) Month of record as an ordered factor (1 = January, 12 = December)

**month\_day** (integer) Full day of the month

**week\_day** (factor) Full day of the week as an ordered factor (1 = Sunday, 7 = Saturday)

**hour** (integer) The hour of the day in 24 hour format

**sensor\_id** (integer) the id of the sensor

**sensor\_name** (character) the full name of the sensor

## Source

<https://data.melbourne.vic.gov.au/Transport-Movement/Pedestrian-volume-updated-monthly-b2ak-trbp>

## Examples

```
## Not run:  
# explore the missingness with vis_miss  
library(naniar)  
  
vis_miss(pedestrian)  
  
# Look at the missingness in the variables  
miss_var_summary(pedestrian)  
  
# There is only missingness in hourly_counts
```

```
# Look at the missingness over a rolling window
library(ggplot2)
gg_miss_span(pedestrian, hourly_counts, span_every = 3000)

## End(Not run)
```

---

prop\_complete      *Return the proportion of complete values*

---

### Description

The complement to prop\_miss

### Usage

```
prop_complete(x)
```

### Arguments

x                      vector or data.frame

### Value

numeric proportion of complete values

### Examples

```
prop_complete(airquality)
prop_complete(airquality$Ozone)
```

---

prop\_complete\_row      *Return a vector of the proportion of missing values in each row*

---

### Description

Substitute for rowMeans(!is.na(data)), but it also checks if input is NULL or is a dataframe

### Usage

```
prop_complete_row(data)
```

### Arguments

data                    a dataframe



**Value**

numeric vector of the proportion of missing values in each row

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#)  
[miss\\_var\\_table](#) [n\\_complete](#) [n\\_complete\\_row](#) [n\\_miss](#) [n\\_miss\\_row](#) [pct\\_complete](#) [pct\\_miss](#)  
[prop\\_complete](#) [prop\\_complete\\_row](#) [prop\\_miss](#)

**Examples**

```
prop_complete_row(airquality)
prop_complete_row(pedestrian)
```

---

prop_miss	<i>Return the proportion of missing values</i>
-----------	--

---

**Description**

This is shorthand for `mean(is.na(x))`

**Usage**

```
prop_miss(x)
```

**Arguments**

x                      vector or data.frame

**Value**

numeric the proportion of missing values in x

**Examples**

```
prop_miss(airquality)
prop_miss(airquality$Ozone)
```

---

prop_miss_row	<i>Return a vector of the proportion of missing values in each row</i>
---------------	--

---

**Description**

Substitute for `rowMeans(is.na(data))`, but it also checks if input is NULL or is a dataframe

**Usage**

```
prop_miss_row(data)
```

**Arguments**

data            a dataframe

**Value**

numeric vector of the proportion of missing values in each row

**See Also**

[miss\\_case\\_pct](#) [miss\\_case\\_prop](#) [miss\\_prop\\_summary\(\)](#) [miss\\_case\\_summary](#) [miss\\_case\\_table](#)  
[miss\\_summary](#) [miss\\_var\\_pct](#) [miss\\_var\\_prop](#) [miss\\_var\\_run](#) [miss\\_var\\_span](#) [miss\\_var\\_summary](#)  
[miss\\_var\\_table](#) [n\\_complete](#) [n\\_complete\\_row](#) [n\\_miss](#) [n\\_miss\\_row](#) [pct\\_complete](#) [pct\\_miss](#)  
[prop\\_complete](#) [prop\\_complete\\_row](#) [prop\\_miss](#)

**Examples**

```
prop_miss_row(airquality)  
prop_miss_row(pedestrian)
```

---

replace_to_na	<i>Replace values with missings</i>
---------------	-------------------------------------

---

**Description**

This function is deprecated, please see [replace\\_with\\_na\(\)](#).

**Usage**

```
replace_to_na(data, to_na = list(), ...)
```

**Arguments**

data            A data.frame  
 to\_na          A named list given the NA to replace values  
 ...            additional arguments for methods.

**Value**

values replaced by NA

---

replace\_with\_na      *Replace values with missings*

---

**Description**

Specify variables and their values that you want to convert to missing values. This is a complement to `tidyr::replace_na`.

**Usage**

```
replace_with_na(data, replace = list(), ...)
```

**Arguments**

data            A data.frame  
 replace        A named list given the NA to replace values for each column  
 ...            additional arguments for methods. Currently unused

**Value**

Dataframe with values replaced by NA.

**See Also**

[replace\\_with\\_na](#) [replace\\_with\\_na\\_all](#) [replace\\_with\\_na\\_at](#) [replace\\_with\\_na\\_if](#)

**Examples**

```
dat_ms <- tibble::tribble(~x, ~y, ~z,
  1, "A", -100,
  3, "N/A", -99,
  NA, NA, -98,
  -99, "E", -101,
  -98, "F", -1)

replace_with_na(dat_ms,
  replace = list(x = -99))
```

```

replace_with_na(dat_ms,
  replace = list(x = -98))

replace_with_na(dat_ms,
  replace = list(x = c(-99, -98)))

replace_with_na(dat_ms,
  replace = list(x = c(-99, -98),
    y = c("N/A")))

replace_with_na(dat_ms,
  replace = list(x = c(-99, -98),
    y = c("N/A"),
    z = c(-101)))

```

---

replace\_with\_na\_all *Replace all values with NA where a certain condition is met*

---

### Description

This function takes a dataframe and replaces all values that meet the condition specified as an NA value, following a special syntax.

### Usage

```
replace_with_na_all(data, condition)
```

### Arguments

data	A dataframe
condition	A condition required to be TRUE to set NA. Here, the condition is specified with a formula, following the syntax: <code>~.x {condition}</code> . For example, writing <code>~.x &lt; 20</code> would mean "where a variable value is less than 20, replace with NA".

### Examples

```

dat_ms <- tibble::tribble(~x, ~y, ~z,
  1, "A", -100,
  3, "N/A", -99,
  NA, NA, -98,
  -99, "E", -101,
  -98, "F", -1)

dat_ms
#replace all instances of -99 with NA
replace_with_na_all(data = dat_ms,
  condition = ~.x == -99)

```

```

# replace all instances of -98 with NA
replace_with_na_all(data = dat_ms,
                    condition = ~.x == -98)

# replace all instances of -99 or -98 with NA
replace_with_na_all(dat_ms,
                    condition = ~.x %in% c(-99, -98))

# replace all instances of -99 or -98, or "N/A" with NA
replace_with_na_all(dat_ms,
                    condition = ~.x %in% c(-99, -98, "N/A"))

# where works with functions
replace_with_na_all(airquality, ~ sqrt(.x) < 5)

```

---

replace_with_na_at	<i>Replace specified variables with NA where a certain condition is met</i>
--------------------	---

---

## Description

Replace specified variables with NA where a certain condition is met

## Usage

```
replace_with_na_at(data, .vars, condition)
```

## Arguments

data	dataframe
.vars	The variables to refer to
condition	A condition required to be TRUE to set NA. Here, the condition is specified with a formula, following the syntax: <code>~.x {condition}</code> . For example, writing <code>~.x &lt; 20</code> would mean "where a variable value is less than 20, replace with NA".

## Value

a dataframe

## Examples

```

dat_ms <- tibble::tribble(~x, ~y, ~z,
  1, "A", -100,
  3, "N/A", -99,
  NA, NA, -98,
  -99, "E", -101,
  -98, "F", -1)

```

```
dat_ms

replace_with_na_at(data = dat_ms,
  .vars = "x",
  condition = ~.x == -99)

replace_with_na_at(data = dat_ms,
  .vars = c("x", "z"),
  condition = ~.x == -99)
```

---

replace\_with\_na\_if     *Replace values with NA based on some condition, for variables that meet some predicate*

---

### Description

Replace values with NA based on some condition, for variables that meet some predicate

### Usage

```
replace_with_na_if(data, .predicate, condition)
```

### Arguments

data	Dataframe
.predicate	A predicate function to be applied to the columns or a logical vector.
condition	A condition required to be TRUE to set NA. Here, the condition is specified with a formula, following the syntax: <code>~.x {condition}</code> . For example, writing <code>~.x &lt; 20</code> would mean "where a variable value is less than 20, replace with NA".

### Value

Dataframe

### Examples

```
dat_ms <- tibble::tribble(~x, ~y, ~z,
  1, "A", -100,
  3, "N/A", -99,
  NA, NA, -98,
  -99, "E", -101,
  -98, "F", -1)

dat_ms

replace_with_na_if(data = dat_ms,
```

```

      .predicate = is.character,
      condition = ~.x == "N/A")

replace_with_na(dat_ms,
  to_na = list(x = c(-99, -98),
              y = c("N/A"),
              z = c(-101)))

```

---

riskfactors	<i>The Behavioral Risk Factor Surveillance System (BRFSS) Survey Data, 2009.</i>
-------------	--

---

### Description

The data is a subset of the 2009 survey from BRFSS, an ongoing data collection program designed to measure behavioral risk factors for the adult population (18 years of age or older) living in households.

### Usage

```
data(riskfactors)
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 245 rows and 34 columns.

### Source

[https://www.cdc.gov/brfss/annual\\_data/annual\\_2009.htm](https://www.cdc.gov/brfss/annual_data/annual_2009.htm)

### See Also

the codebook: [http://ftp.cdc.gov/pub/data/brfss/codebook\\_09.rtf](http://ftp.cdc.gov/pub/data/brfss/codebook_09.rtf)

Format: a data frame with 245 observations on the following 34 variables.

`state` A factor with 52 levels. The labels and states corresponding to the labels are as follows:  
 1:Alabama, 2:Alaska, 4:Arizona, 5:Arkansas, 6:California,8:Colorado, 9:Connecticut, 10:Delaware,  
 11:District of Columbia,12:Florida, 13:Georgia, 15:Hawaii, 16:Idaho, 1 :Illinois,18:Indiana,  
 19:Iowa, 20:Kansas, 21:Kentucky, 22:Louisiana,23:Maine, 24:Maryland, 25:Massachusetts,  
 26:Michigan,27:Minnesota, 28:Mississippi, 2:Missouri, 30:Montana,31:Nebraska, 32:Nevada,  
 33:New Hampshire, 34:New Jersey, 35:NewMexico, 36:New York, 37:North Carolina, 38:North  
 Dakota, 39:Ohio,40:Oklahoma, 41:Oregon, 42:Pennsylvania, 44:Rhode Island, 45:SouthCarolina,  
 46:South Dakota, 47:Tennessee, 48:Texas, 49:Utah, 50:Vermont, 51:Virginia, 53:Washington,  
 54:West Virginia,55:Wisconsin, 56:Wyoming, 66:Guam, 72:Puerto Rico, 78:Virgin Islands

`sex` A factor with levels Male Female.

`age` A numeric vector from 7 to 97.

weight\_lbs The weight without shoes in pounds.

height\_inch The weight without shoes in inches.

bmi Body Mass Index (BMI). Computed by weight in Kilogram /(height in Meters \* height in Meters). Missing if any of weight or height is missing.

marital A factor with levels Married Divorced Widowed Separated NeverMarried UnmarriedCouple.

pregnant Whether pregnant now with two levels Yes and No.

children A numeric vector giving the number of children less than 18 years of age in household.

education A factor with the education levels 1 2 3 4 5 6 as 1: Never attended school or only kindergarten; 2: Grades 1 through 8 (Elementary); 3: Grades 9 through 11 (Some high school); 4: Grade 12 or GED (High school graduate); 5: College 1 year to 3 years (Some college or technical school); 6: College 4 years or more (College graduate).

employment A factor showing the employment status with levels 1 2 3 4 5 7 8. The labels mean – 1: Employed for wages; 2: Self-employed; 3: Out of work for more than 1 year; 4: Out of work for less that 1 year; 5: A homemaker; 6: A student; 7:Retired; 8: Unable to work.

income The annual household income from all sources with levels <10k 10-15k 15-20k 20-25k 25-35k 35-50k 50-75k >75k Dontknow Refused.

veteran A factor with levels 1 2 3 4 5. The question for this variable is: Have you ever served on active duty in the United States Armed Forces, either in the regular military or in a National Guard or military reserve unit? Active duty does not include training for the Reserves or National Guard, but DOES include activation, for example, for the Persian Gulf War. And the labels are meaning: 1: Yes, now on active duty; 2: Yes, on active duty during the last 12 months, but not now; 3: Yes, on active duty in the past, but not during the last 12 months; 4: No, training for Reserves or National Guard only; 5: No, never served in the military.

hispanic A factor with levels Yes No corresponding to the question: are you Hispanic or Latino?

health\_general Answer to question "in general your health is" with levels Excellent VeryGood Good Fair Poor Refused.

health\_physical The number of days during the last 30 days that the respondent's physical health was not good. -7 is for "Don't know/Not sure", and -9 is for "Refused".

health\_mental The number of days during the last 30 days that the respondent's mental health was not good. -7 is for "Don't know/Not sure", and -9 is for "Refused".

health\_poor The number of days during the last 30 days that poor physical or mental health keep the respondent from doing usual activities, such as self-care, work, or recreation. -7 is for "Don't know/Not sure", and -9 is for "Refused".

health\_cover Whether having any kind of health care coverage, including health insurance, pre-paid plans such as HMOs, or government plans such as Medicare. The answer has two levels: Yes and No.

provide\_care Whether providing any such care or assistance to a friend or family member during the past month, with levels Yes and No.

activity\_limited Whether being limited in any way in any activities because of physical, mental, or emotional problems, with levels Yes and No.

drink\_any Whether having had at least one drink of any alcoholic beverage such as beer, wine, a malt beverage or liquor during the past 30 days, with levels Yes and No.



drink\_days The number of days during the past 30 days that the respondent had at least one drink of any alcoholic beverage. -7 is for "Don't know/Not sure", and -9 is for "Refused".

drink\_avg The number of drinks on the average the respondent had on the days when he/she drank, during the past 30 days. -7 is for "Don't know/Not sure", and -9 is for "Refused".

smoke\_100 Whether having smoked at least 100 cigarettes in the entire life, with levels Yes and No.

smoke\_days The frequency of days now smoking, with levels Everyday Somedays and NotAtAll(not at all).

smoke\_stop Whether having stopped smoking for one day or longer during the past 12 months because the respondent was trying to quit smoking, with levels Yes and No.

smoke\_last A factor with levels 3 4 5 6 7 8 corresponding to the question: how long has it been since last smoking cigarettes regularly? The labels mean: 3: Within the past 6 months (3 months but less than 6 months ago); 4: Within the past year (6 months but less than 1 year ago); 5: Within the past 5 years (1 year but less than 5 years ago); 6: Within the past 10 years (5 years but less than 10 years ago); 7: 10 years or more; 8: Never smoked regularly.

diet\_fruit The number of fruit the respondent eat every year, not counting juice. -7 is for "Don't know/Not sure", and -9 is for "Refused".

diet\_salad The number of servings of green salad the respondent eat every year. -7 is for "Don't know/Not sure", and -9 is for "Refused".

diet\_potato The number of servings of potatoes, not including french fries, fried potatoes, or potato chips, that the respondent eat every year. -7 is for "Don't know/Not sure", and -9 is for "Refused".

diet\_carrot The number of carrots the respondent eat every year. -7 is for "Don't know/Not sure", and -9 is for "Refused".

diet\_vegetable The number of servings of vegetables the respondent eat every year, not counting carrots, potatoes, or salad. -7 is for "Don't know/Not sure", and -9 is for "Refused".

diet\_juice The number of fruit juices such as orange, grapefruit, or tomato that the respondent drink every year. -7 is for "Don't know/Not sure", and -9 is for "Refused".

library(MissingDataGUI) (named brfss)

## Examples

```

# explore the missingness with vis_miss
library(naniar)

vis_miss(riskfactors)

# Look at the missingness in the variables
miss_var_summary(riskfactors)

# and now as a plot
gg_miss_var(riskfactors)

# Look at the missingness in bmi and poor health
library(ggplot2)

```

```
p <-  
ggplot(riskfactors,  
       aes(x = health_poor,  
           y = bmi)) +  
  geom_miss_point()  
  
p  
  
# for each sex?  
p + facet_wrap(~sex)  
# for each education bracket?  
p + facet_wrap(~education)
```

---

shadow\_shift

*Shift missing values to facilitate missing data exploration/visualisation*

---

## Description

shadow\_shift transforms missing values to facilitate visualisation, and has different behaviour for different types of variables. For numeric variables, the values are shifted to 10 variable plus some jittered noise, to separate repeated values, so that missing values can be visualised along with the rest of the data.

## Usage

```
shadow_shift(x, ...)
```

## Arguments

x	a variable of interest to shift
...	extra arguments to pass

## See Also

[add\\_shadow\\_shift\(\)](#) [cast\\_shadow\\_shift\(\)](#) [cast\\_shadow\\_shift\\_label\(\)](#)

## Examples

```
airquality$Ozone  
shadow_shift(airquality$Ozone)  
library(dplyr)  
airquality %>%  
  mutate(Ozone_shift = shadow_shift(Ozone))
```

---

stat_miss_point	<i>stat_miss_point</i>
-----------------	------------------------

---

### Description

stat\_miss\_point adds a geometry for displaying missingness to geom\_point

### Usage

```
stat_miss_point(mapping = NULL, data = NULL, geom = "point",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)
```

### Arguments

mapping	Set of aesthetic mappings created by <code>ggplot2::aes()</code> or <code>ggplot2::aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), is combined with the default mapping at the top level of the plot. You only need to supply mapping if there isn't a mapping defined for the plot.
data	A data frame. If specified, overrides the default data frame defined at the top level of the plot.
geom,	stat Override the default connection between <code>geom_point</code> and <code>stat_point</code> .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function
na.rm	If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders</code> .
...	other arguments passed on to <code>ggplot2::layer()</code> . There are three types of arguments you can use here: <ul style="list-style-type: none"> <li>• Aesthetics: to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code>.</li> <li>• Other arguments to the layer, for example you override the default stat associated with the layer.</li> <li>• Other arguments passed on to the stat.</li> </ul>

test\_if\_dataframe      *Test if input is a data.frame*

---

**Description**

Test if input is a data.frame

**Usage**

```
test_if_dataframe(x)
```

**Arguments**

x                    object

**Value**

an error if input (x) is a data.frame

**Examples**

```
## Not run:  
# success  
test_if_dataframe(airquality)  
#fail  
my_test <- matrix(10)  
test_if_dataframe(my_test)  
  
## End(Not run)
```

---

test\_if\_missing      *Test if the input is Missing*

---

**Description**

Test if the input is Missing

**Usage**

```
test_if_missing(x)
```

**Arguments**

x                    object

**Value**

an error if input (x) is not specified

**Examples**

```
## Not run:  
# success  
my_test <- x  
test_if_null(my_test)  
#fail  
test_if_missing()  
  
## End(Not run)
```

---

test_if_null	<i>Test if the input is NULL</i>
--------------	----------------------------------

---

**Description**

Test if the input is NULL

**Usage**

```
test_if_null(x)
```

**Arguments**

x                    object

**Value**

an error if input (x) is NULL

**Examples**

```
## Not run:  
# success  
test_if_null(airquality)  
#fail  
my_test <- NULL  
test_if_null(my_test)  
  
## End(Not run)
```

---

where_na	<i>Which rows and cols contain missings?</i>
----------	--

---

**Description**

Internal function that is short for `which(is.na(x), arr.ind = TRUE)`. Creates array index locations of missing values in a dataframe.

**Usage**

```
where_na(x)
```

**Arguments**

x                    a dataframe

**Value**

a matrix with columns "row" and "col", which refer to the row and column that identify the position of a missing value in a dataframe

**See Also**

[which\\_na\(\)](#)

**Examples**

```
where_na(airquality)
where_na(oceanbuoys$sea_temp_c)
```

---

which_na	<i>Which elements contain missings?</i>
----------	---

---

**Description**

Equivalent to `which(is.na())` - returns integer locations of missing values.

**Usage**

```
which_na(x)
```

**Arguments**

x                    a dataframe

**Value**

integer locations of missing values.

**See Also**

[where\\_na\(\)](#)

**Examples**

```
which_na(airquality)
```

---

which\_var\_na

*Which variables contain missing values?*

---

**Description**

It can be helpful when writing other functions to just return the names of the variables that contain missing values.

**Usage**

```
which_var_na(data)
```

**Arguments**

data            a data.frame

**Value**

character vector of variable names

**Examples**

```
## Not run:  
which_var_na(airquality)  
  
which_var_na(iris)  
  
## End(Not run)
```

# Index

## \*Topic **datasets**

- GeomMissPoint, 18
  - oceanbuoys, 44
  - pedestrian, 47
  - riskfactors, 55
- add\_any\_miss, 4
- add\_any\_miss(), 4–10, 15–17, 27
- add\_label\_missings, 5
- add\_label\_missings(), 4–10, 15–17, 27
- add\_label\_shadow, 5
- add\_label\_shadow(), 4–10, 15–17, 27
- add\_miss\_cluster, 6
- add\_miss\_cluster(), 4–10, 15–17, 27
- add\_n\_miss, 7
- add\_n\_miss(), 4–6, 9, 10, 27
- add\_prop\_miss, 8
- add\_prop\_miss(), 4–10, 15–17, 27
- add\_shadow, 9
- add\_shadow\_shift, 10
- add\_shadow\_shift(), 4–10, 15–17, 27, 58
- add\_span\_counter, 10
- all\_row\_complete, 11
- all\_row\_miss, 12
- any\_row\_miss, 12
- as\_shadow, 13
- as\_shadow.data.frame, 13
- bind\_shadow, 14
- bind\_shadow(), 4–10, 15–17, 27
- cast\_shadow, 15
- cast\_shadow(), 4–10, 27
- cast\_shadow\_shift, 16
- cast\_shadow\_shift(), 15–17, 58
- cast\_shadow\_shift\_label, 16
- cast\_shadow\_shift\_label(), 15–17, 58
- gather\_shadow, 17
- geom\_miss\_point, 18
- geom\_miss\_point(), 20–25
- GeomMissPoint, 18
- gg\_miss\_case, 20
- gg\_miss\_case(), 21–25
- gg\_miss\_case\_cumsum, 20, 21, 22–25
- gg\_miss\_fct, 22
- gg\_miss\_fct(), 20, 21, 23–25
- gg\_miss\_span, 22
- gg\_miss\_span(), 20–22, 24, 25
- gg\_miss\_var, 23
- gg\_miss\_var(), 20–25
- gg\_miss\_var\_cumsum, 24
- gg\_miss\_var\_cumsum(), 20–25
- gg\_miss\_which, 25
- gg\_miss\_which(), 20–25
- ggplot2::aes(), 19, 59
- ggplot2::aes\_(), 19, 59
- ggplot2::layer(), 19, 59
- group\_by\_fun, 26
- label\_miss\_1d, 27
- label\_miss\_2d, 28
- label\_missings, 26
- label\_shadow, 28
- label\_shadow\_matrix, 29
- miss\_case\_pct, 29
- miss\_case\_prop, 30
- miss\_case\_summary, 31
- miss\_case\_table, 32
- miss\_prop\_summary, 32
- miss\_prop\_summary(), 30–40, 42, 43, 49, 50
- miss\_scan\_count, 33
- miss\_summary, 34
- miss\_var\_pct, 35
- miss\_var\_prop, 36
- miss\_var\_run, 36
- miss\_var\_span, 38
- miss\_var\_summary, 39
- miss\_var\_table, 40



n\_complete, 41  
n\_complete\_row, 42  
n\_miss, 42  
n\_miss\_row, 43  
naniar, 40  
naniar-ggproto (GeomMissPoint), 18  
naniar-package (naniar), 40  
  
oceanbuoys, 44  
  
pct\_complete, 45  
pct\_miss, 46  
pedestrian, 47  
prop\_complete, 48  
prop\_complete\_row, 48  
prop\_miss, 49  
prop\_miss\_row, 50  
  
replace\_to\_na, 50  
replace\_with\_na, 51  
replace\_with\_na(), 50  
replace\_with\_na\_all, 52  
replace\_with\_na\_at, 53  
replace\_with\_na\_if, 54  
riskfactors, 55  
  
shadow\_shift, 58  
stat\_miss\_point, 59  
StatMissPoint (GeomMissPoint), 18  
  
test\_if\_dataframe, 60  
test\_if\_missing, 60  
test\_if\_null, 61  
  
where\_na, 62  
where\_na(), 63  
which\_na, 62  
which\_na(), 62  
which\_var\_na, 63